Geant Exascale Pilot Project

Philippe Canal for the Geant Exascale Pilot team.



Outline

2

Exascale: GPU for the foreseeable future

- ECP: Investing in software for Exascale
- Geant On GPU: prior efforts
- Exascale Geant Pilot: a collaboration between ECP and HEP
- Goals and strategy





Exascale: GPU for the Foreseeable Future

Perlmutter (NERSC-9) 2020: AMD and Nvidia GPUs

- GPU nodes will have a GPU to CPU ratio of 4:1
- 256GB memory per node or greater
- >4000 node CPU partition, approximately same capability as full Cori system today
- Still under NDA

3

- **Frontier** (ORNL) 2021 and **El Capitan** (LLNL) 2023:
 - one AMD Epyc processor to four AMD Instinct graphics cards
 - A21 (ANL) 2021:
 - Intel Configurable Spatial Accelerator
 - Dataflow graph engine
 - Maps compiler execution graph (IR) to hardware fabric
 - Elements of modern switches, FPGAs, KNL
 - Intel GPU, Intel ONE API







Philippe Canal - G4 R&D Working Group - May 2019

Exascale Computing Project



- Preparing for the next generation of supercomputers.
- 10-year project led by six DOE and NNSA laboratories and executed in collaboration with academia and industry
- Accelerating delivery of a capable exascale computing ecosystem for breakthroughs in scientific discovery, energy assurance, economic competitiveness, and national security.
- Goal to deliver breakthrough modeling and simulation solutions that analyze more data in less time, providing insights and answers to the most critical US challenges in scientific discovery, energy assurance, economic competitiveness, and national security

Geant On GPU: Prior Efforts

MPEXS-DNA: simulator for track structures and radiation chemistry at subcellular scale.

- EM Physics and diffusion and chemical reactions for molecular species for micro dosimetry simulation
- Speed up 2900x (NVidia Volta)

GATE simulation on GPU

- Simulations of Preclinical and Clinical Scans in Emission Tomography, Transmission Tomography and Radiation Therapy
- GPU only for voxelized phantom (very time consuming) 60x
- GPU only for PET and CT applications (validated in Bert et al2013) 70x
- G4CU: Geant4 Based Simulation of Radiation Dosimetry in CUDA
 - Focused on dose calculation for radiation therapy
 - EM Physics in H₂O in voxel geometry, 75~97 speedup
- Opticks: GPU Optical photon simulation for Particle Physics with NVDIA's OptiX
 - 200x speedup with mobile GPU (GeForce GT 750M)
- Applications where unlike HEP there are usually a small subset of code/work/configuration that dominates execution time and that have inherently less conditional branches
- Those packages are often closed source.

Exascale Geant Pilot: a Collaboration between ECP and HEP

- First (AFAIK) official collaboration between HEP and ECP (FNAL, ORNL and LBL)
- In the context of the ECP co-design focus area
- To study and characterize architecture and performance
- Will encapsulate, at first, a minimum set of physics and software framework processes necessary to describe a HEP simulation problem.
- Will then be used to exercise/research communication, computation and data access patterns.
- Main objective is to identify re-engineering opportunities that will increase event throughput by improving single node performance and being able to make efficient use of of the next generation of accelerators available in exascale facilities.

Goals and Strategies

- Investigate how to best use GPUs for full HEP simulation
 - Estimate higher limit of speed-ups.
- Explore memory access, computation ordering and CPU/GPU communication patterns
 - Require flexibility on data structure and code structure
- Avoid constraints (backward compatibility, tracking change 'upstream')
 - Reuse or leverage existing packages
 - newer components like VecGeom, GeantV magnetic field and physics implementation
- Focus on NVidia compiler at first
 - Will need to understand AMD and Intel programming direction
 - Likely might switch to a library like Kokkos.
- Avoid over-simplification
 - Look at the whole Geant simulation chain not just a few components.

GPUs and Polymorphism

Detector simulation relies (needs) polymorphism to support a wide range of particles and physics models and to allow customization of the simulation (eg. Physics lists).

Copying objects to and from GPU invalidates virtual table (if any)

Static polymorphism leads to templating of large part of code template <typename ParticuleType> class Track; template <typename TrackType> class TrackManager;

etc...

Partial Static Polymorphism

Type aware container

9

template <typename... ParticleTypes> struct VariadicTrackManager {

// without template param, we push to generic at end
void PushTrack(Track* track) { m_tracks[num_types]->PushTrack(track); }

// without template params, we pop from end
Track* PopTrack()

template <typename ParticleType> TrackCaster<ParticleType>* PopTrack();

Virtuality "Eraser"

10

Mechanism to access content of concrete class 'Track' without going through virtual table:

template <typename Type> struct TrackCaster : public Track

Type* GetParticleDefinition() const { return static_cast<Type*>(m_pdef); } std::string GetParticleName() const

return static_cast<Type*>(m_pdef)->Type:::GetName();

// no need to reimplement GetTrackId()

Philippe Canal - G4 R&D Working Group - May 2019

};

Dispatching

Configuration of actions at compile time via "traits".

template <typename... ParticleTypes> struct component_tuple

```
// ...
void AlongStep( ... )
```

```
// component::start is the operator struct this is the "TypeList"
using ProcessTypes = std::tuple< AlongStep<ParticleTypes>... >;
```

```
// "ProcessTypes" (our operator typelist) expands to
```

```
//
```

11

// std::tuple < AlongStep < Photon >,

```
/ AlongStep < Electron > >;
```

apply<void>::access< ProcessTypes >(....);

// ...

Separation Of State and Access.

- Split the struct holding the Track class from the accessors and modifiers functions.
- Allow for smaller set of interfaces
 - Geometry info, Physics info, Etc.
- Will allow change/investigation in physical layout of Track transparently to the rest of code.
 - Possibly different layout (eg. shorter/smaller) on GPU than CPU
 - Struct of Array vs Array of Struct
 - Etc.
- Allow to factor out more 'track' related functionality
 - while still keep it close to the (kind of) code that uses it and without having unwieldly large classes.
- "Should" disappear thanks to inlining and optimization

13

New projects to explore GPU use for HEP Geant Simulations.

Targeting the upcoming Exascale class machine

Collaboration between ECP and HEP involving FNAL, ORNL and LBL
 Joint expertise in HPC, GPU, Geant4/V

Currently setting up the infrastructure classes and functionality.

Backup slides

Exascale Geant Pilot: a collaboration between ECP and HEP

- Daniel Elvira, Krzysztof Genser, Panagiotis Spentzouris (FNAL)
- Philippe Canal, Soon Yun, Guilherme Lima (FNAL)
- Paolo Calafiura, Phillip Colella, Zach Marshall (LBL)
- Jonathan Madsen, Yunsong Wang (LBL)
- Seth Johnson, TBA (ORNL)
- Tom Evans (ORNL)

ECP Strategic Goals and Outcomes



Goal: Applications

16

Foundational element of ECP and vehicle for delivery of results from the exascale systems enabled by ECP. Each addresses an exascale challenge problem—a problem of strategic importance and national interest that is intractable without at least 50 times the computational power of today's systems

Goal: Software Technologies

Underlying technologies that applications are built and relied on: essential for application performance, portability, integrity, and resilience. Spans low-level system software to high-level application development environments, including infrastructure for large-scale data science and an expanded and vertically integrated software stack with advanced mathematical libs and frameworks, extreme-scale programming environments, tools, and visualization libs

Goal: Hørdware and Integration

- Partnerships between U.S. vendors and DOE's application and software developers to develop a new generation of commodity computing components. Assure at least two diverse and viable exascale computing technology pathways for the Nation to meet identified mission needs and proactively engage and integrate with DOE HPC facilities in the process.
- **Outcome: Accelerated delivery of a capable exascale computing ecosystem** to provide breakthrough solutions addressing our most critical challenges in scientific discovery, energy assurance, economic competitiveness, and national security
 - Capable: Wide range of applications can effectively use the systems developed through ECP, ensuring that both science and security needs will be addressed (affordable, usable, useful)
 - Exascale: Enhance application performance by 50x relative to today's systems (e.g., ORNL's Titan)
 - Ecosystem: Not just more powerful systems, but all methods and tools needed for effective use of ECP-enabled exascale systems to be acquired by DOE labs

Application Challenges on Exascale Systems



- Get ready by running well on today's machines
 - If you can run well on today's machines you will be ready enough that you can update your code in time
 - Leverage standards or frameworks:

17

- OpenMP, Kokkos and RAJA will likely work on whatever comes out and can limit your exposure to porting yourself
- Avoid the newest language features unless needed
 - In particular, modern Fortran support with OpenMP tends to lag behind

Exascale is happening through increased parallelism not clock speeds

- While a continuation of the last decade this makes two things harder
 - Improving time to solution
 - Non-parallel code sections/algorithms
- This will be made worse because network latencies, gpu kernel launch times and other Amdahl bottlenecks are hard to drive down at the rate of throughput increases.
- Total memory capacity is unlikely to scale

Ian Karlin , LLNL at 2019 ECP Annual Meeting

Philippe Canal - G4 R&D Working Group - May 2019

2019-05-08

Most significant challenges in preparing applications for exascale machines



- Granularity of parallelism
 - Expose a lot, use less.
- Transition overhead and data movement
 - Network and memory speed will decrease relative to compute speed.
- Data-structure restructuring needed Not just loops

Hal Finkel, ANL at 2019 annual ECP meeting

Programming Preparation Thoughts



- Do what you can on existing machines
 - Port to GPUs now (E.g., Summit, Sierra)
 - Port to KNL now (vectorization) (E.g., Cori)

- David Donofrio, BNL at 2019 ECP Annual Meeting
- Begin to partition Data structures and multi-level problem decomposition
- Accelerators (surprise!) are important
 - Highest fraction of peak performance is here
 - SW tools continue to improve
- Increasing on-node parallelism
- Interconnect performance not tracking increase in node performance
- More memory per node (good news!)
 - Less memory per thread (not so good news)
- Trending to tighter integration of CPUs and accelerators
- Increasing coherence simplifies programming... but often at performance cost
- Memory / Allocation increasingly important

ECP focus areas

20

Application Development

Develops and enhances predictive capability of applications critical to DOE across science, energy, and national security mission space

- Targeted development of requirements-based models, algorithms, and methods
- Integration of appropriate software and hardware via codesign methodologies
- Systematic improvement of exascale system readiness and utilization
- Demonstration and assessment of effective software integration

Software Technology (ST)

Spans low-level operational software to high-level applications software development environments, including support for data needs of science and national security activities at exascale

- Line of sight to applications
- Software Development Kits (SDKs) to enhance the drive for collaboration
- Delivery of specific software products

Hardware and Integration (HI)

Ensures integrated delivery of specific outcomes and products on targeted systems at leading DOE HPC facilities

- PathForward
- Hardware evaluation
- Application integration at facilities
- Software deployment at facilities
- Facility resource utilization
- Training and productivity



Energy Applications Modeling and simulation of existing and future technologies for the efficient and 22 responsible production of energy that meets the growing needs of the US Modeling of plasma-based particle reactors, leading to the design WarpX of a 1 TeV electron-positron high-energy collider Increase the efficiency and reduce the cost and time to market for complex MFIX-Exa multiphase flow reactor designs for carbon capture and storage technologies Design optimization for small modular reactors, a key component ExaSMR in affordable nuclear energy for the future Whole-device, high-fidelity modeling approach for magnetically confined fusion WDMApp plasmas applicable to a high-performance advanced tokamak regime Detailed modeling and simulation of combustion physics to reduce petroleum use Combustion-(increase efficiency) and reduce greenhouse gas emissions PELE ExaWind Modeling and simulation of whole wind plant performance

ExaSMR – motivation for HPC in nuclear reactor modeling

- Value of HPC computing in nuclear was demonstrated by CASL
 - 2012: Shift (MC) neutronics calculations were used to validate startup conditions of the new WEC AP1000[®] reactor
 - 2015 2017: VERA Shift is being used to model WB2 startup
 - These calculations use most of Titan
 - 18 688 compute nodes
 - 10¹²/MC particle histories
 - CASI is primarily focused on industry-level computing
 - Not addressing coupled, multicycle high-resolution modeling
 - High-order neutronics used for benchmarking
 - Transport for Ex-core dosimetry and pressure vessel fluence



Small Modular Reactor (SMR) Challenge Problems require high-resolution models

- Coupled MC neutronics/CFD simulations steady state operation
 - Model movable absorbers
 - Optimize fuel assembly designs with strong power gradients
- Coupled MC neutronics/CFD simulations of the full core in steady-state, low-flow critical heat flux (CHF) conditions
 - Demonstrate a substantial steady-state CHF margin
- Transient coupled MC neutronics/CFD simulations of the low-flow natural circulation startup
 - Significant advancement over empirically correlated 5 × 5 rod bundle experiments that are used for design

Couple MC neutronics and LES, multiphase CFD are required to address these challenges

Required compute capability for Challenge Problems



Geant on GPU

26

- MPEXS-DNA: https://www.ncbi.nlm.nih.gov/pubmed/30593679 , https://aapm.onlinelibrary.wiley.com/doi/full/10.1002/mp.13048
- Gate http://www.opengatecollaboration.org/sites/default/files/Talk6.pdf , https://www.creatis.insa-lyon.fr/~dsarrut/articles/Bert2012.pdf , http://www.opengatecollaboration.org/#
- G4CU:

https://kds.kek.jp/indico/event/15926/session/30/contribution/110/material/slides/0.pdf

Opticks:

https://indico.cern.ch/event/647154/contributions/2733123/attachments/1529047/239224 5/opticks_gpu_optical_photon_simulation_sep2017_wollongong.pdf